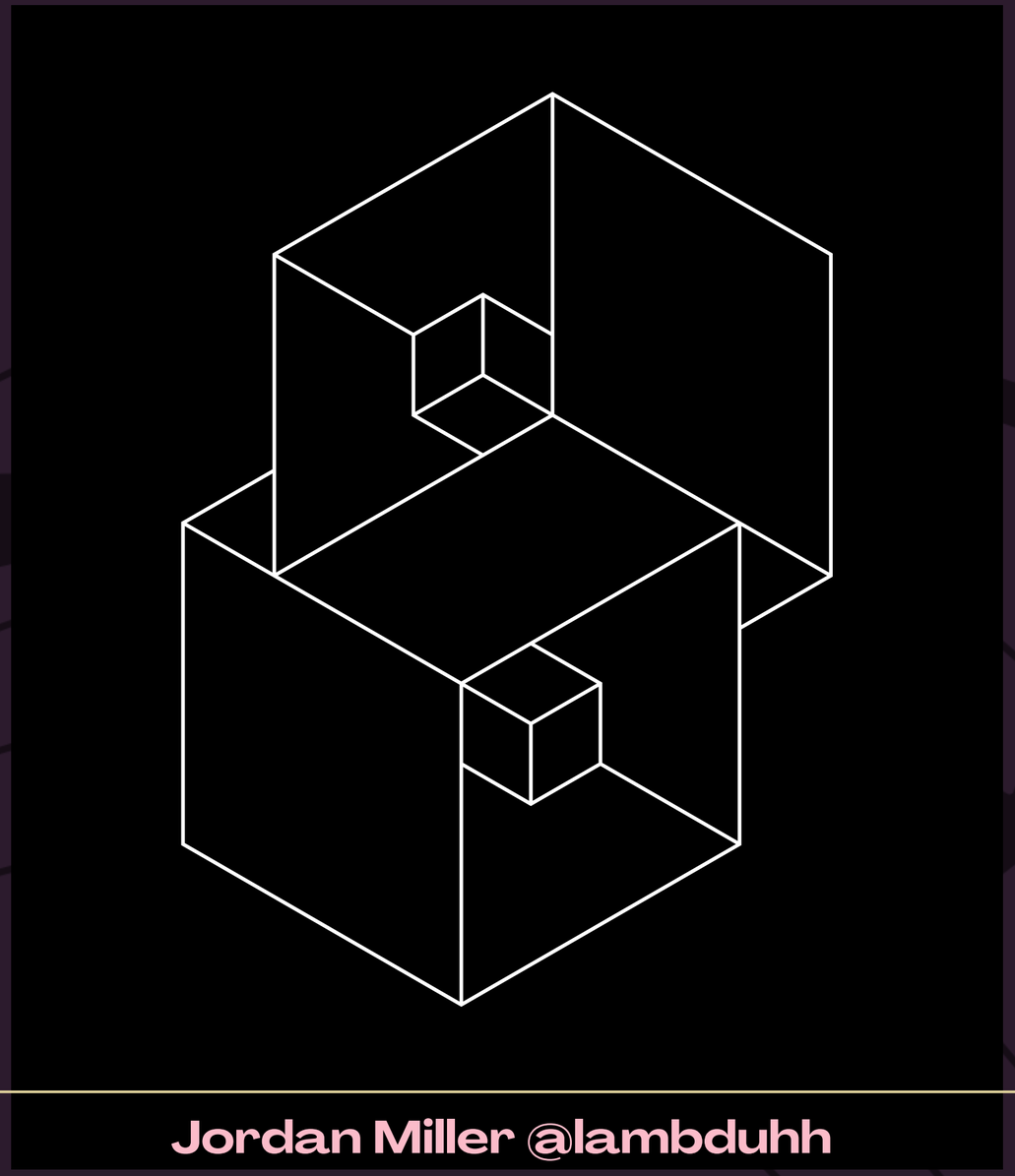
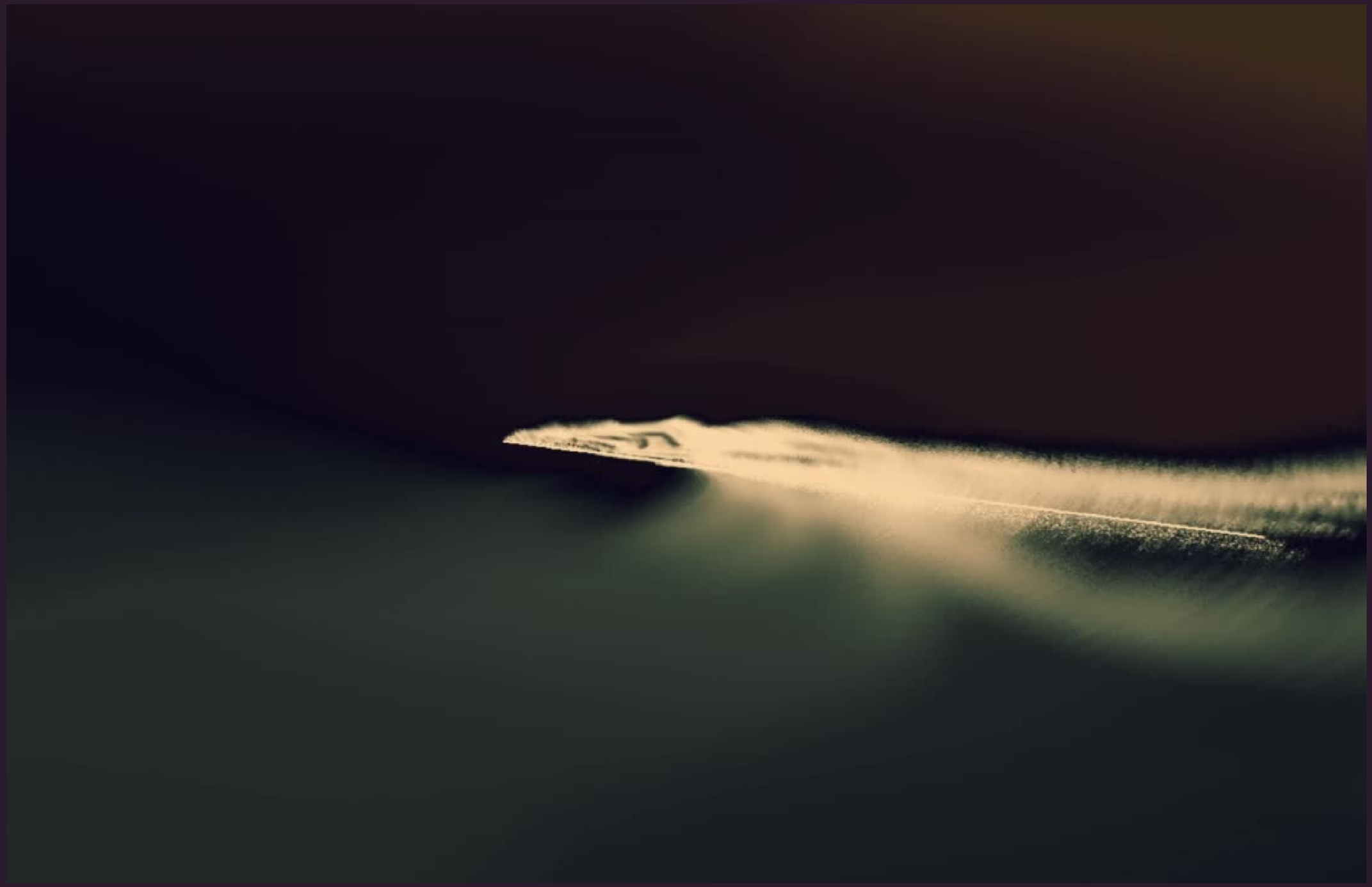
Find The write rhythm for your software composition



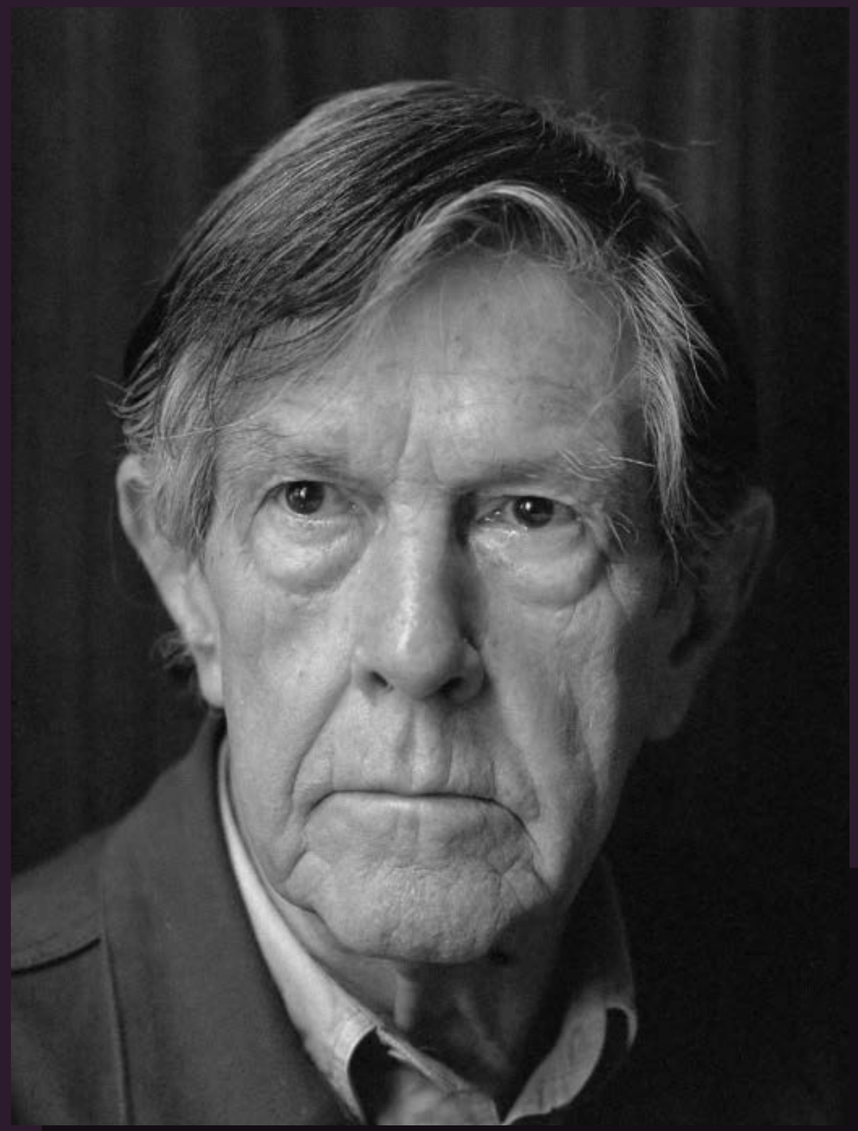
Jordan Miller @lambduhh

Lambda Days 2023



Do you recognize this?

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144

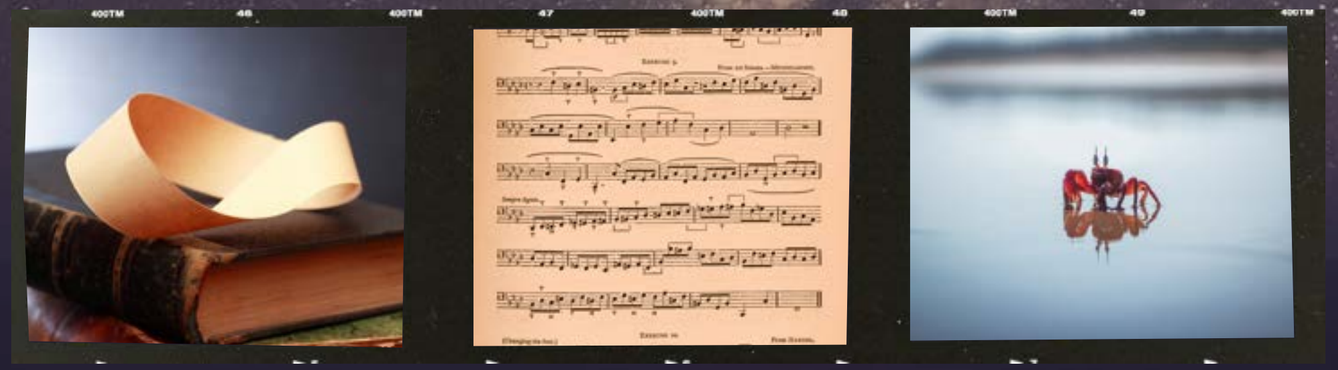


”I have nothing to say,
I am saying it,
and that is poetry.”

JOHN CAGE

(1912–1992)

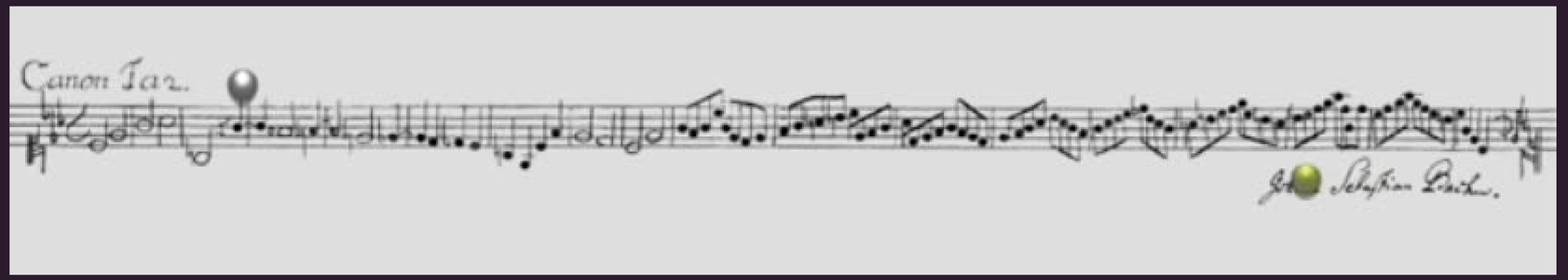
Do you recognize this?

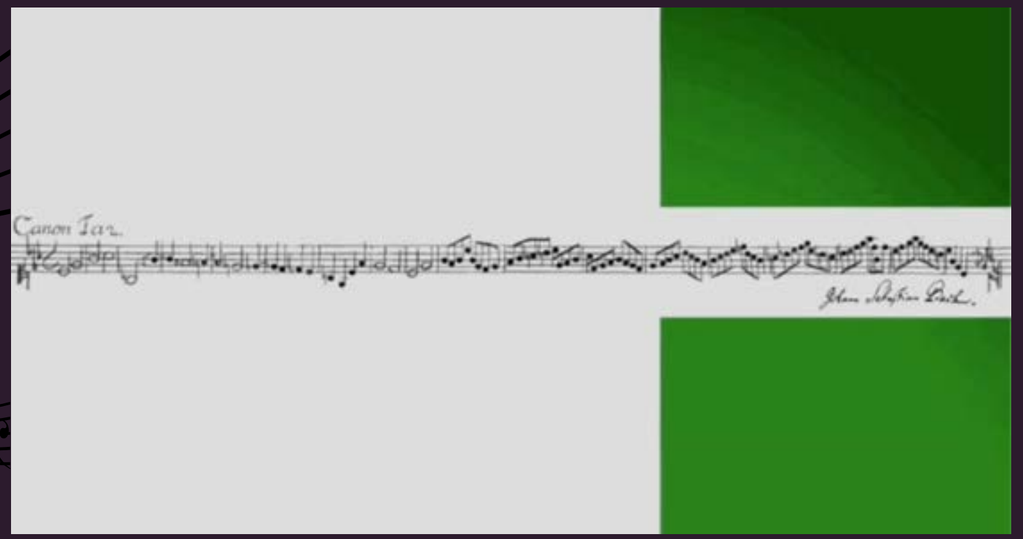


0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144

JS BACH

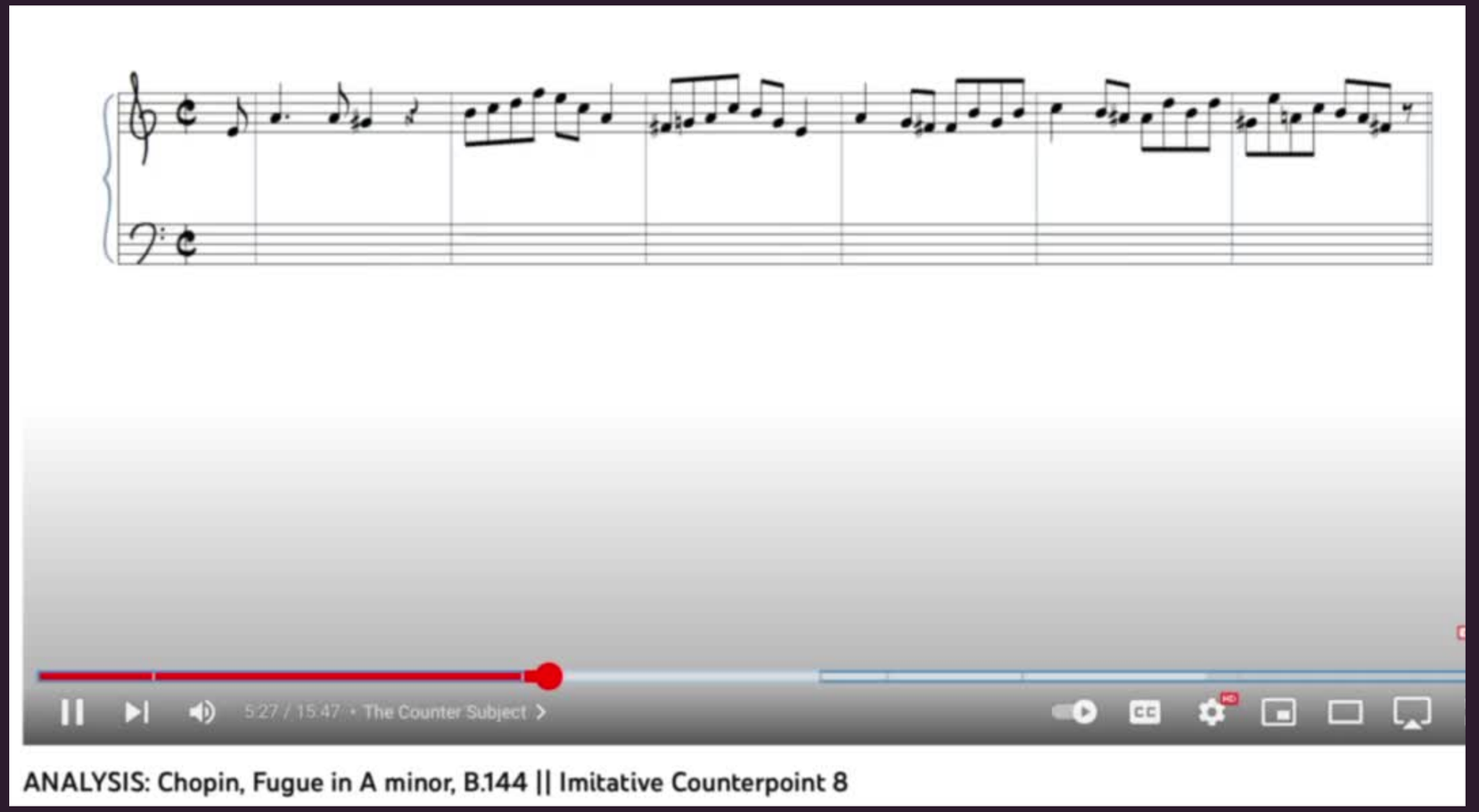
(AKA "OLD BACH")





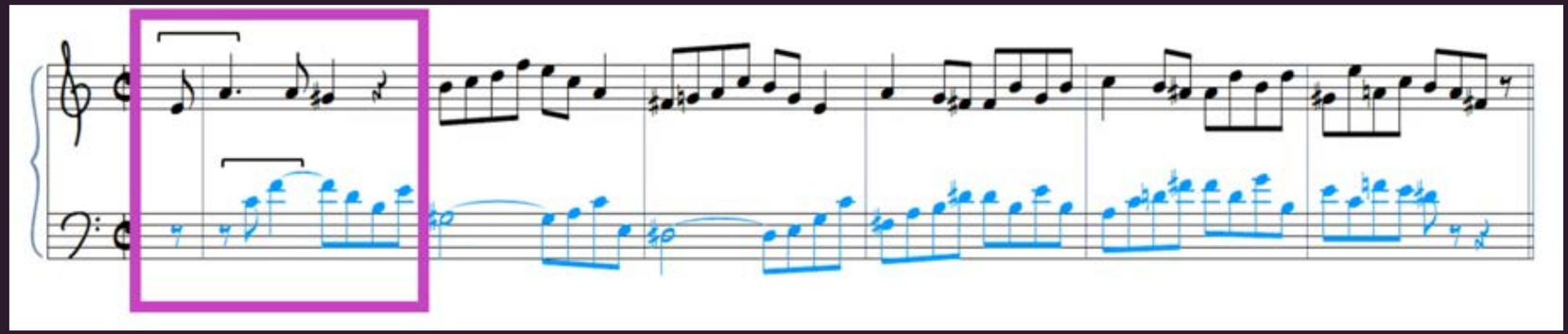
FRÉDÉRIC CHOPIN

(1810-1849)



FRÉDÉRIC CHOPIN

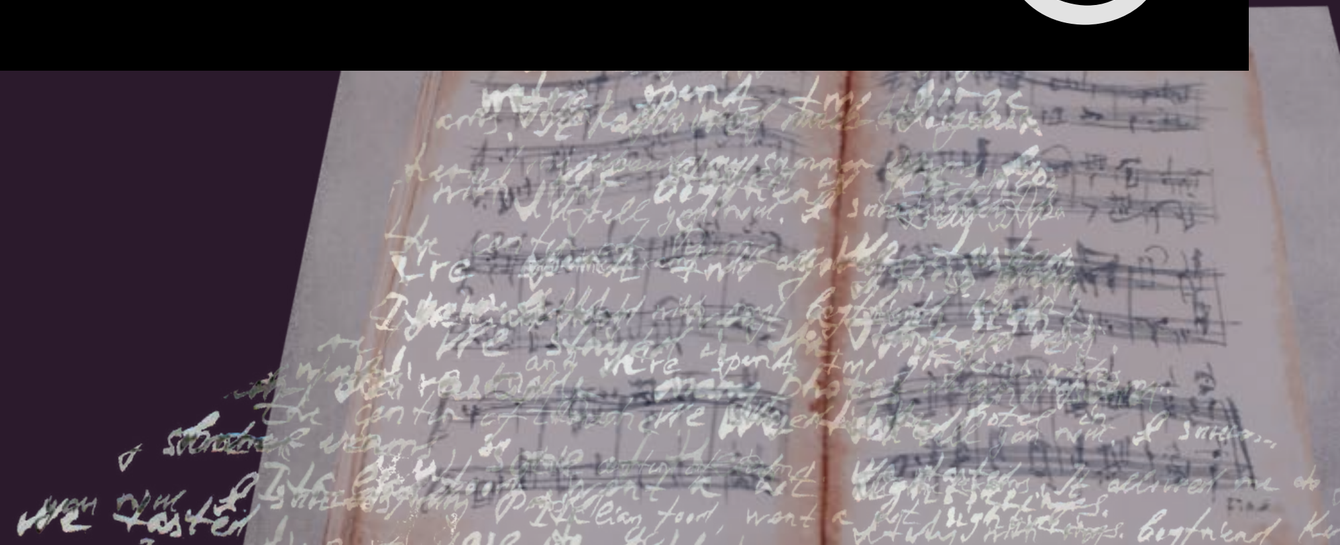
(1810-1849)



Once upon a Time... at Vouch

UH OH, DOCUMENTATION WOES

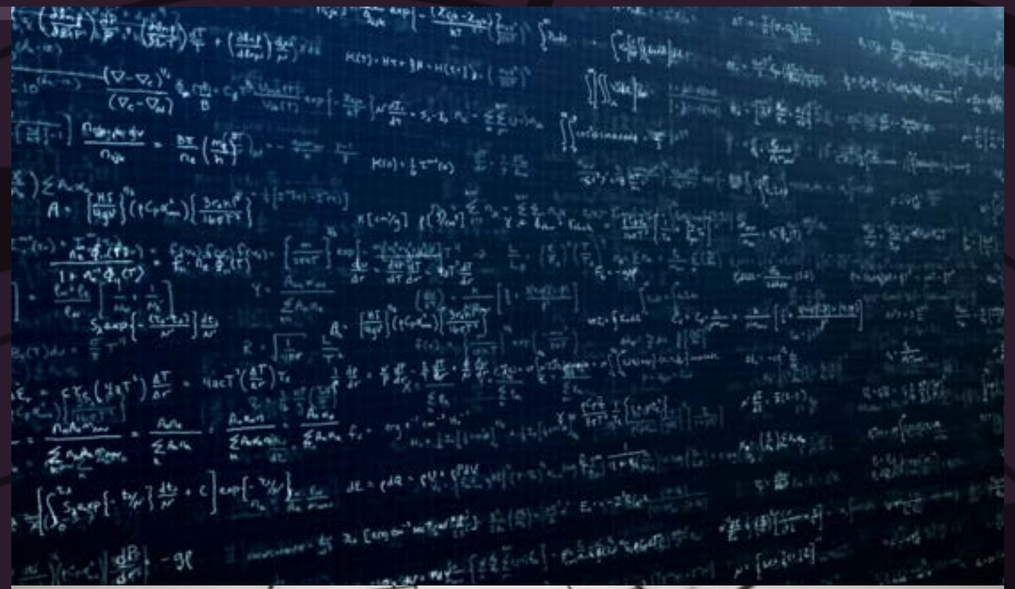
UH OH. DOCUMENTATION WOES

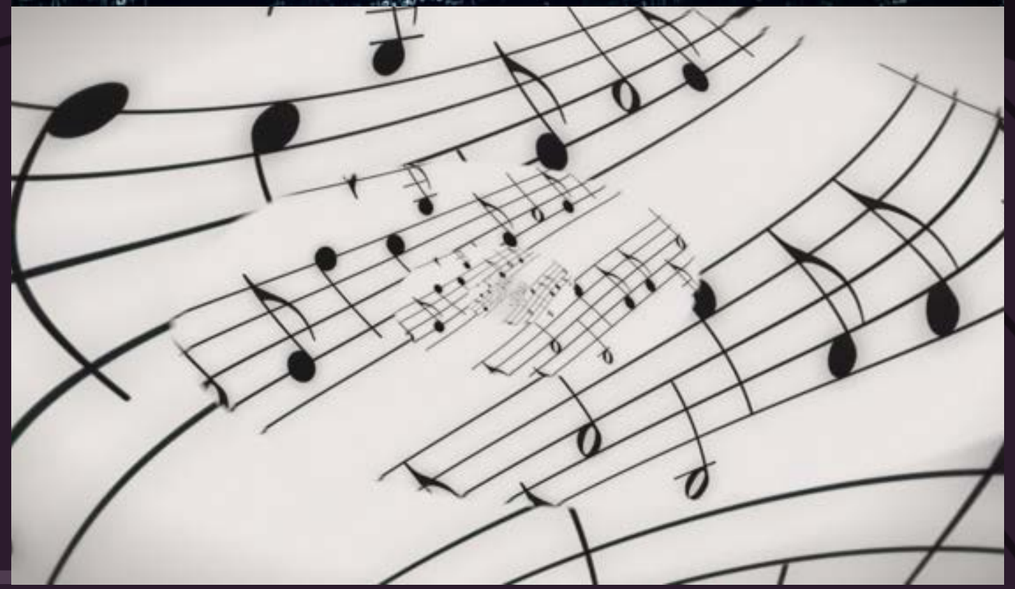



What is a Composer?

Music Theory

- How music works,
  - TL;DR it's all just math






400TM
46
47
400TM
48
400TM
49
400TM
34
21
5
3
1
2
8
13
0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144

Do you recognize this?

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144


CHAOS
INTELLIGENCE
PATTERNS

Intelligence loves patterns

```clojure
;; Define the subject as list of notes
(def subject ["A" "D" "E" "A"])

;; Define generic procedure with multi arity options
(defmulti play (λ [pattern transformation &args] transformation))

;; Returns subject, does not modify
(defmethod play :default [_ _] subject)

```

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144

Intelligence loves patterns

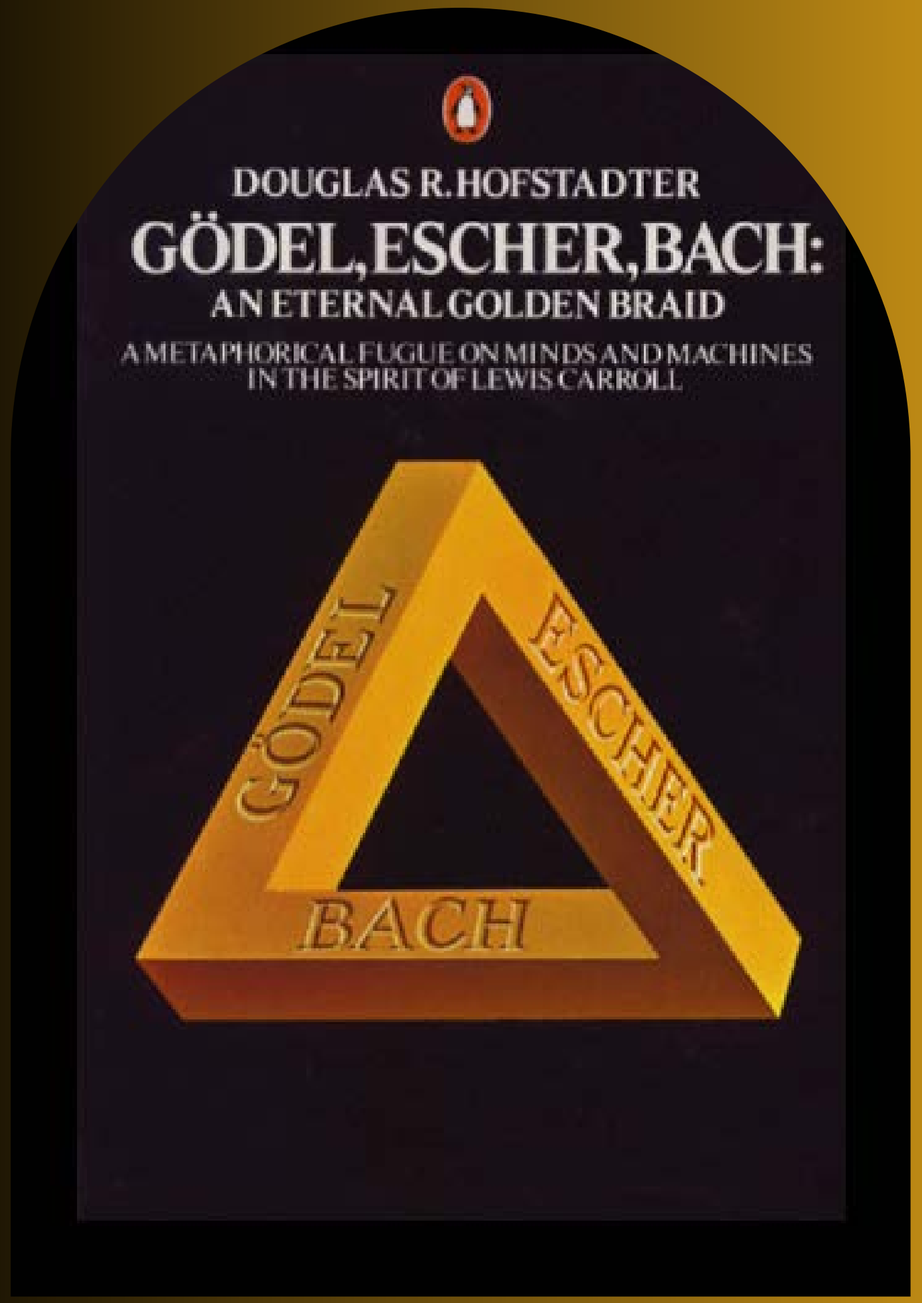
```clojure
;; Define the subject as list of notes
(def subject ["A" "D" "E" "A"])

;; Define generic procedure with multi arity options
(defmulti play (λ [pattern transformation &args] transformation))

;; Returns subject, does not modify
(defmethod play :default [_ _] subject)

;; Reverses subject
(defmethod play :retrograde [_ _] (reverse subject))

;; Changes the key
(defmethod play :transpose-minor [_ _] (map #(str % "m") subject))
```

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144

Intelligence loves patterns

```clojure
;; Define the subject as list of notes
(def subject ["A" "D" "E" "A"])

;; Define generic procedure with multi arity options
(defmulti play (λ [pattern transformation &args] transformation))

;; Returns subject, does not modify
(defmethod play :default [_ _] subject)

;; Reverses subject
(defmethod play :retrograde [_ _] (reverse subject))

;; Changes the key
(defmethod play :transpose-minor [_ _] (map #(str % "m") subject))

;; Creates crab canon
(defmethod play :mobius [_ _]
  (let [half (quot (count subject) 2)        ;; find half of the subject
        first-half (subvec subject 0 half)   ;; create list of the first half
        second-half (reverse (subvec subject half))] ;; create list of second half, reverse
    (concat first-half second-half first-half))) ;; combine into continous mobius strip

```

```clojure
;; Define the subject as list of notes
(def subject ["A" "D" "E" "A"])

;; Define generic procedure with multi arity options
(defmulti play (λ [pattern transformation &args] transformation))

;; Returns subject, does not modify
(defmethod play :default [_ _] subject)

;; Reverses subject
(defmethod play :retrograde [_ _] (reverse subject))

;; Changes the key
(defmethod play :transpose-minor [_ _] (map #(str % "m") subject))

;; Creates crab canon
(defmethod play :mobius [_ _]
  (let [half (quot (count subject) 2)       ;; find half of the subject
        first-half (subvec subject 0 half)  ;; create list of the first half
        second-half (reverse (subvec subject half))] ;; create list of second half, reverse
    (concat first-half second-half first-half))) ;; combine into continous mobius strip

(comment

  (play "original" :default)
  ;; => ["A" "D" "E" "A"]

  (play "backwards" :retrograde)
  ;; => ("A" "E" "D" "A")

  (play "to-minor-key" :transpose-minor)
  ;; => ("Am" "Dm" "Em" "Am")

  (play "crab-canon" :mobius)
  ;; => ("A" "D" "A" "E" "A" "D")
```



Chapter VI:

The Location of Meaning

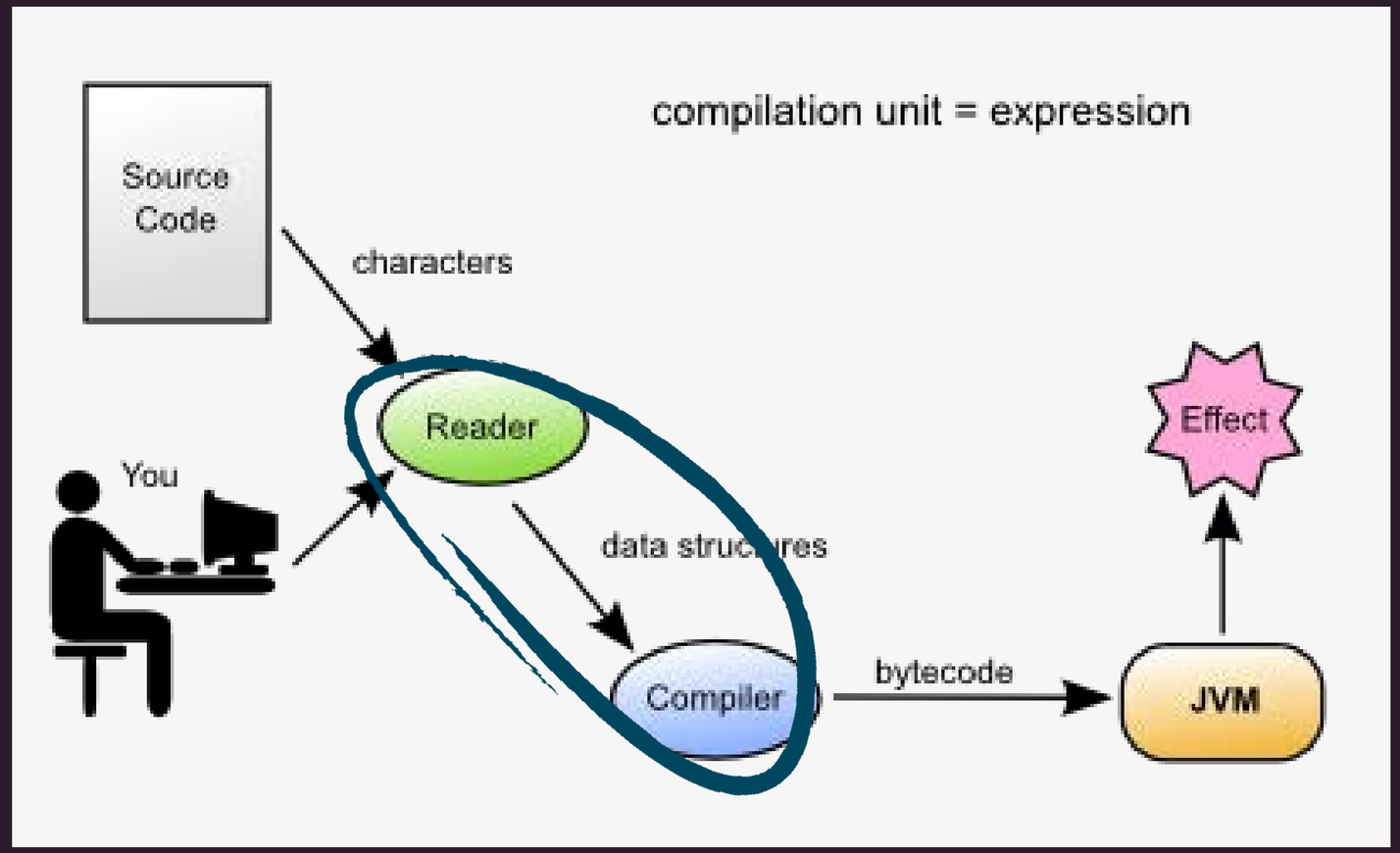
To find LOCATION OF MEANING we must consider 3 LEVELS OF UNDERSTANDING

FRAME | OUTER MESSAGE | INNER MESSAGE

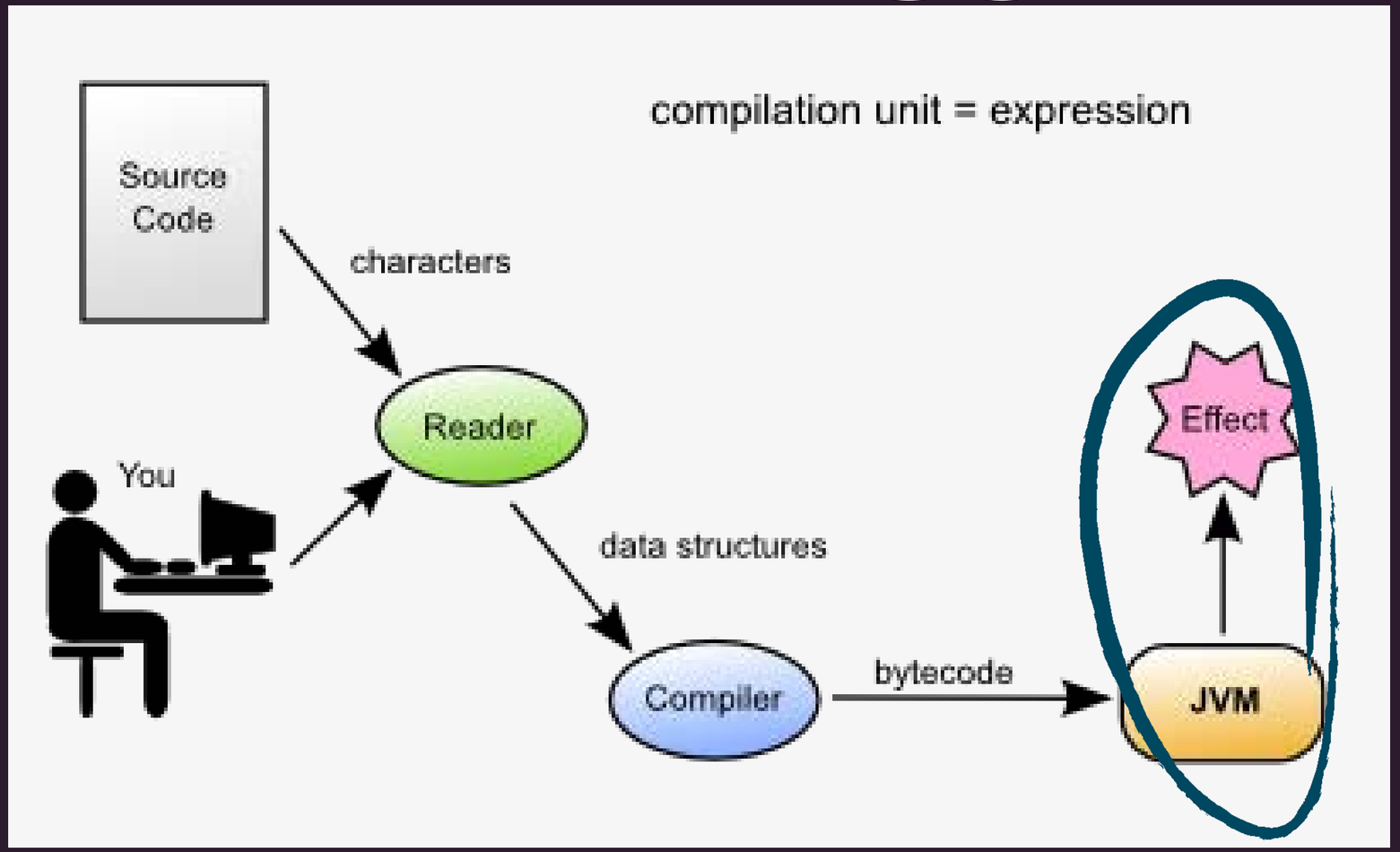
Frame


(define divide-list
(lambda (l)
(let divide
(p '())
(d '()))
(if (null? l)
(define merge
(lambda (s1 s2 . pred?)
(let ((<= (if (null? pred?) <= (car pred?))))
(let merge ((s1 s1)
(s2 s2))
(cond ((null? s1) s2)
((null? s2) s1)
((<= (car s1) (car s2)) (cons (car s1) (merge (cdr s1) s2)))
(else (cons (car s2) (merge s1 (cdr s2))))))))
(define merge-sort
(lambda (l . pred?)
(let ((<= (if (null? pred?) <= (car pred?))))
(if (or (null? l) (null? (cdr l)))
l
(let ((divided-list (divide-list l)))
(merge (merge-sort (car divided-list) <=)
(merge-sort (cdr divided-list) <=))))))

Outer Message



3 Inner message




"VOUCH!"
BACK TO THE STORY

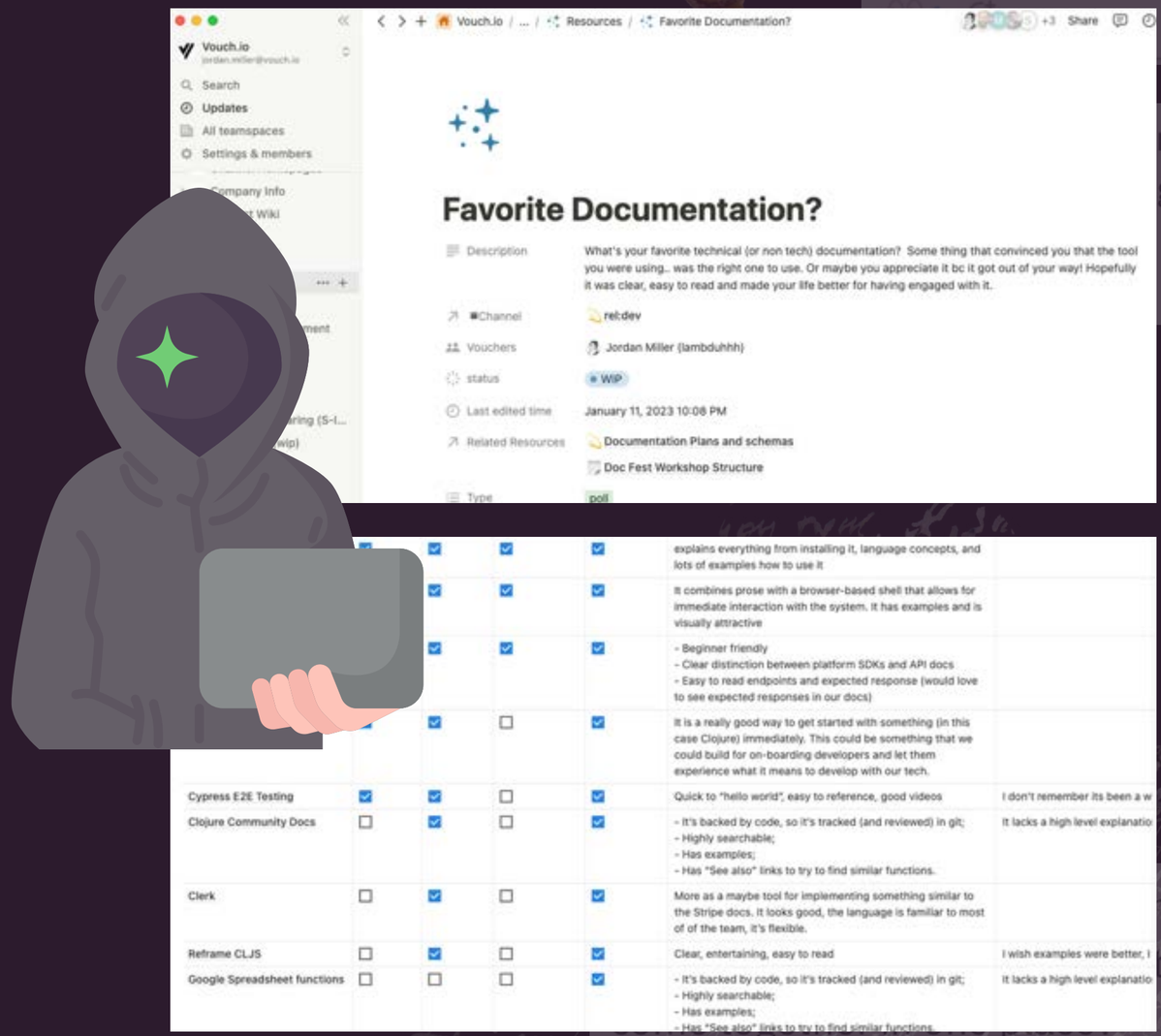
Engineering process and policies don't seem to be written down anywhere?

1

Having documents, tasks, and timelines all in one place (Notion)

1

As the company grows, it'll be harder to keep tabs on what's happening throughout the company and know what others are working on.



	☑	☑	☑	explains everything from installing it, language concepts, and lots of examples how to use it	
	☑	☑	☑	It combines prose with a browser-based shell that allows for immediate interaction with the system. It has examples and is visually attractive	
	☑	☑	☑	- Beginner friendly - Clear distinction between platform SDKs and API docs - Easy to read endpoints and expected response (would love to see expected responses in our docs)	
	☑	☑	☐	☑	It is a really good way to get started with something (in this case Clojure) immediately. This could be something that we could build for on-boarding developers and let them experience what it means to develop with our tech.
Cypress E2E Testing	☑	☑	☐	☑	Quick to "hello world", easy to reference, good videos	I don't remember its been a w
Clojure Community Docs	☐	☑	☐	☑	- It's backed by code, so it's tracked (and reviewed) in git; - Highly searchable; - Has examples; - Has "See also" links to try to find similar functions.	It lacks a high level explanatio
Clerk	☐	☑	☐	☑	More as a maybe tool for implementing something similar to the Stripe docs. It looks good, the language is familiar to most of of the team, it's flexible.	
Reframe CLJS	☐	☑	☐	☑	Clear, entertaining, easy to read	I wish examples were better, i
Google Spreadsheet functions	☐	☐	☐	☑	- It's backed by code, so it's tracked (and reviewed) in git; - Highly searchable; - Has examples; - Has "See also" links to try to find similar functions.	It lacks a high level explanatio

Not enough documentation
work or why decisions we

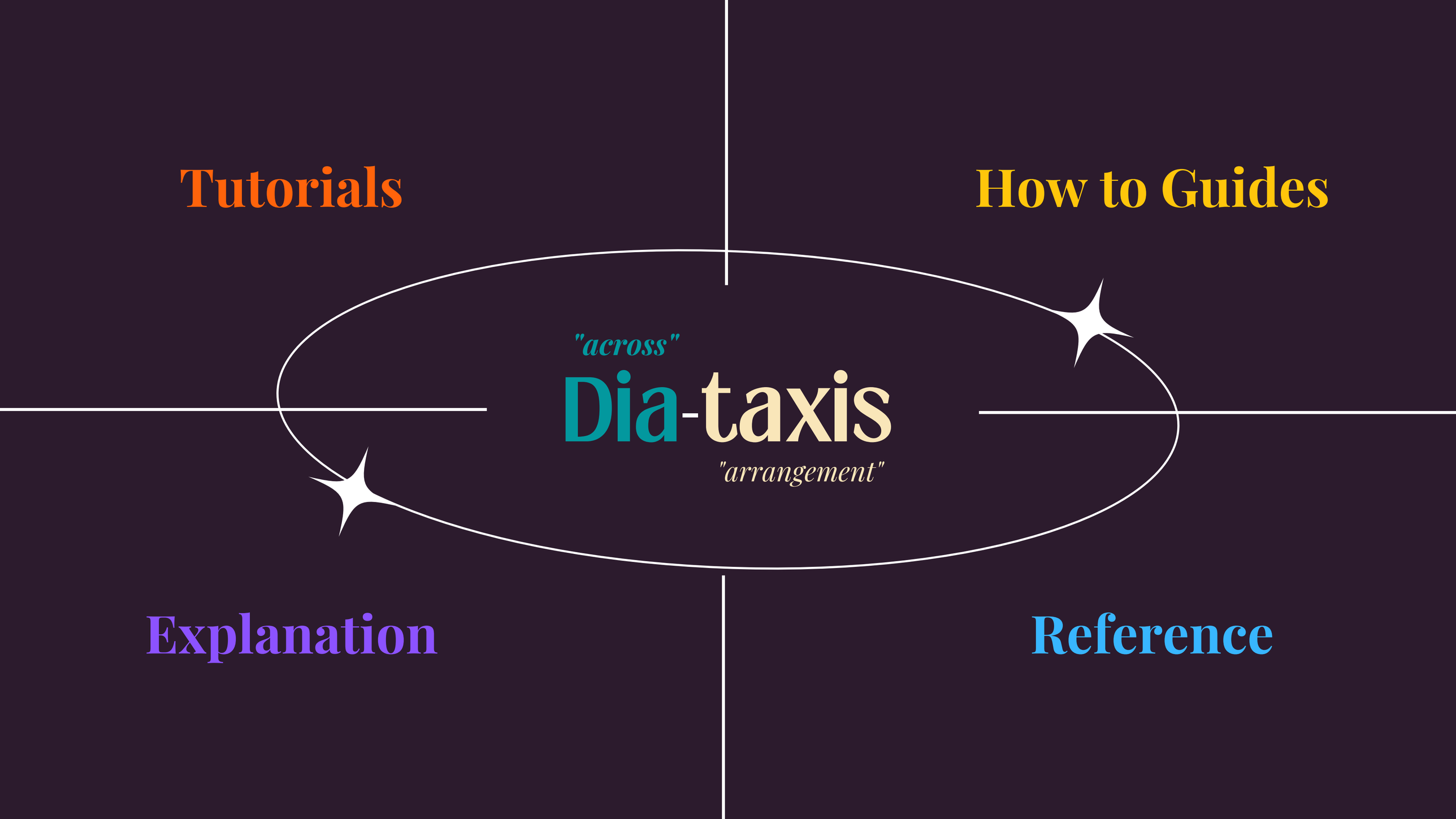
1

inefficient processes arou dge
sharing. I feel like there is time wasted
and unneeded anxiety created as a result

ns documentation
g

ean info is
e and

1


Tutorials
How to Guides
"across"
Dia-taxis
"arrangement"
Explanation
Reference

Tutorials
for learning

How to Guides
for tasks

Explanation
for background

Reference
quick info

Pure Danger Tech

navigation

home

The Miller Principle

11 Jul 2007

I made this up years ago but it holds today as much as ever:

> **The Miller Principle**: No one reads anything.

This principle applies to the following:

- User documentation
- Specifications
- Code comments
- Any text on a user interface
- Any email longer than one line

I had some other principles too but I figured no one would read them…. :)

Missed Connections

- 8 SESSIONS
- 4 HOURS
- 3-6 PEOPLE PER SESSION
- 45 MIN TALK ON DIATAXIS
- 3 GROUP ACTIVITIES
- RECURSIVE IMPROVEMENT VIA FEEDBACK SURVEY

Intention:

We are here to participate in an experience that gives us a **shared understanding** of how to produce high quality technical documentation.

How To Recognize Success?

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144

☑ Shorter Time onboarding

☑ New Artifacts Completed

system.								
How to avoid 401 error code/unauthorized communication between mop bridge and digital-key api	Stable	When creating a new environment, doing rollbacks or simply changing infrastructure, mop bridge needs to have its crypto identity enrolled to be able to comunicate with digital-key api	How to Guide, Documentation	Jordan Miller (lambduhh	Matheus Afinovicz	Build how to avoid mop b	rel:dev, Toyota Delivery, Software, DevOps + Infrastructure	Voucher Internal
How to access datadog logs for deployed environments	Stable	Directs the user in accessing datadog and checking logs for deployes environments.	How to Guide, Documentation		Matheus Afinovicz	Write how to access data	rel:dev, Software, Vouch Key Auto, Dogfooding & Testing, DevOps + Infrastructure	Voucher Internal
How to Init Auth for Software Dev- Configuring Git, AWS, SSH	Stable	Authentication required for a software developer to git set up to allow them to develop vouch apps. (Formerly in "Dev Startup)	Documentation, How to Guide	Heather, Stijn Ophei	Jordan Miller (lambduhhh		Software	Voucher Internal
How to configure Intellij REPL for cljs software dev	Complete	How to configure the REPL using cursive in the Intellij IDE (Formerly in "Dev Startup")	Documentation, How to Guide		Jordan Miller (lambduhhh		Software	Voucher Internal
Tutorial: Change service code without restarting Docker container	Complete	A lesson that contains practical steps and serves our study.	Documentation, Tutorial	Jordan Miller (lambduhh	Bernard Labno	Create Tutorial on how to	rel:dev, Software, Product	Voucher Internal
How to git update all vouch repos	Stable	Ever wanted to pull all your repos at once? check this out.	Documentation, How to Guide, Tips a	Heather, Vinícius Pic	Jordan Miller (lambduhhh		Software	Voucher Internal
How to track errors with stack-trace	WIP	Directs the user in finding the place where errors happening using the stack-trace.	How to Guide, Documentation	Giorgio Rossa	Giorgio Rossa	Write How-To-Guide on S	rel:dev, Software	Voucher Internal
MFikes shares a way to skip zephyr flash	Stale	A discord private discussion brought out to light	Explanation, Documentation		Jordan Miller (lambduhhh		Software, Firmware	
Reference - Nordic Board nRF5340 DK from Digikey, Manufacture Documentation	Stable	Documentation provided by the manufacture of nRF5340 board. Includes component reference diagram, video explainer and PDF download. See links to their documentation portal for further reading.	Reference Guide, Documentation, Arc	Jordan Miller (lambduhh	Jordan Miller (lambduhhh	making HowTo guide for f	rel:dev, Dogfooding & Testing, Hardware, Firmware, Software	Sharable

☑ Features integrated more quickly

feat: implement new transport ble #244

Merged V1pi merged 56 commits into `main` from `new-transport-ble` on Mar 13

Conversation 1 | Commits 56 | Checks 2 | Files changed 14 | +92 −77

Assets / Resources

status	Aa Resource	Description
Stable	Reference - BLE Transports for React Native	BLE Transports for React Native - This document is for anyone who is building a React Native app and wants to use BLE to communicate with AON/FOB (Vouch Devices).
Stable	Reference - BLE Transports	BLE Transports API Reference - This document is for anyone who wants to use BLE to communicate with AON/FOB (Vouch Devices). It is also useful for reference for BLE Developer.

How To DIY?

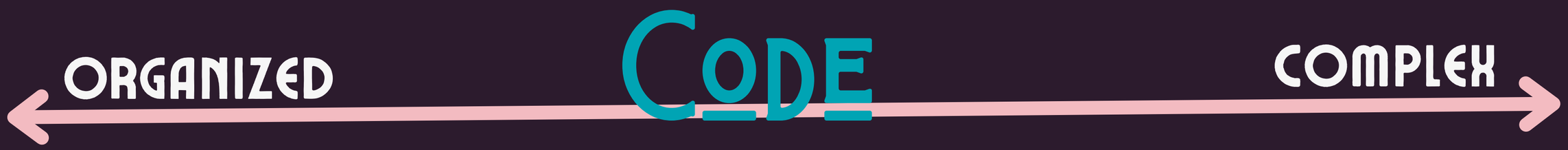
CONSIDER THE OUTER MESSAGE YOU'VE PROVIDED
AND THE INNER MESSAGE YOU ARE
TRYING TO COMMUNICATE

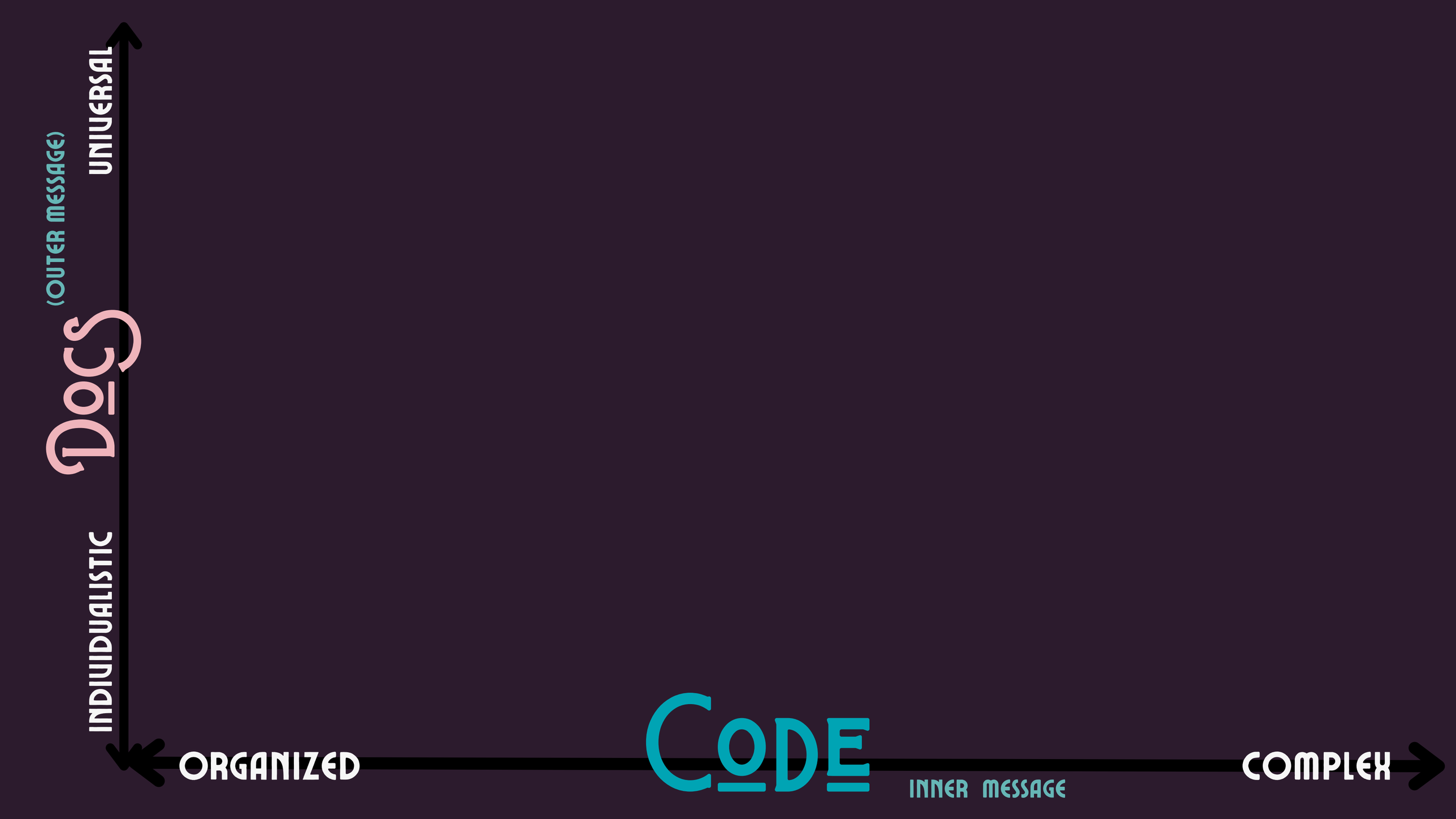
Outer Message

1. Consider your team documentation practices
    a. are your technical artifacts ***on the same page or playing out of tune***?
        i. How good is your **Outer Message**?

Inner Message

1. Consider your code
    a. is it *conventional?*
    organized?
    or *complex?*




UNIVERSAL
(OUTER MESSAGE)
DOCS
INDIVIDUALISTIC
ORGANIZED
CODE
INNER MESSAGE
COMPLEX

UNIVERSAL

DOCS (OUTER MESSAGE)

INDIVIDUALISTIC

Open Source / Unicorn Dev Team

- docs optimized for UX
- kept up to date
- spec/schema
- strong/static typed
- 100% test coverage
- documentation that leverages shared language
- use framework as intended
- extend syntax w doc strings, metadata
- prioritize code readability over fanciness

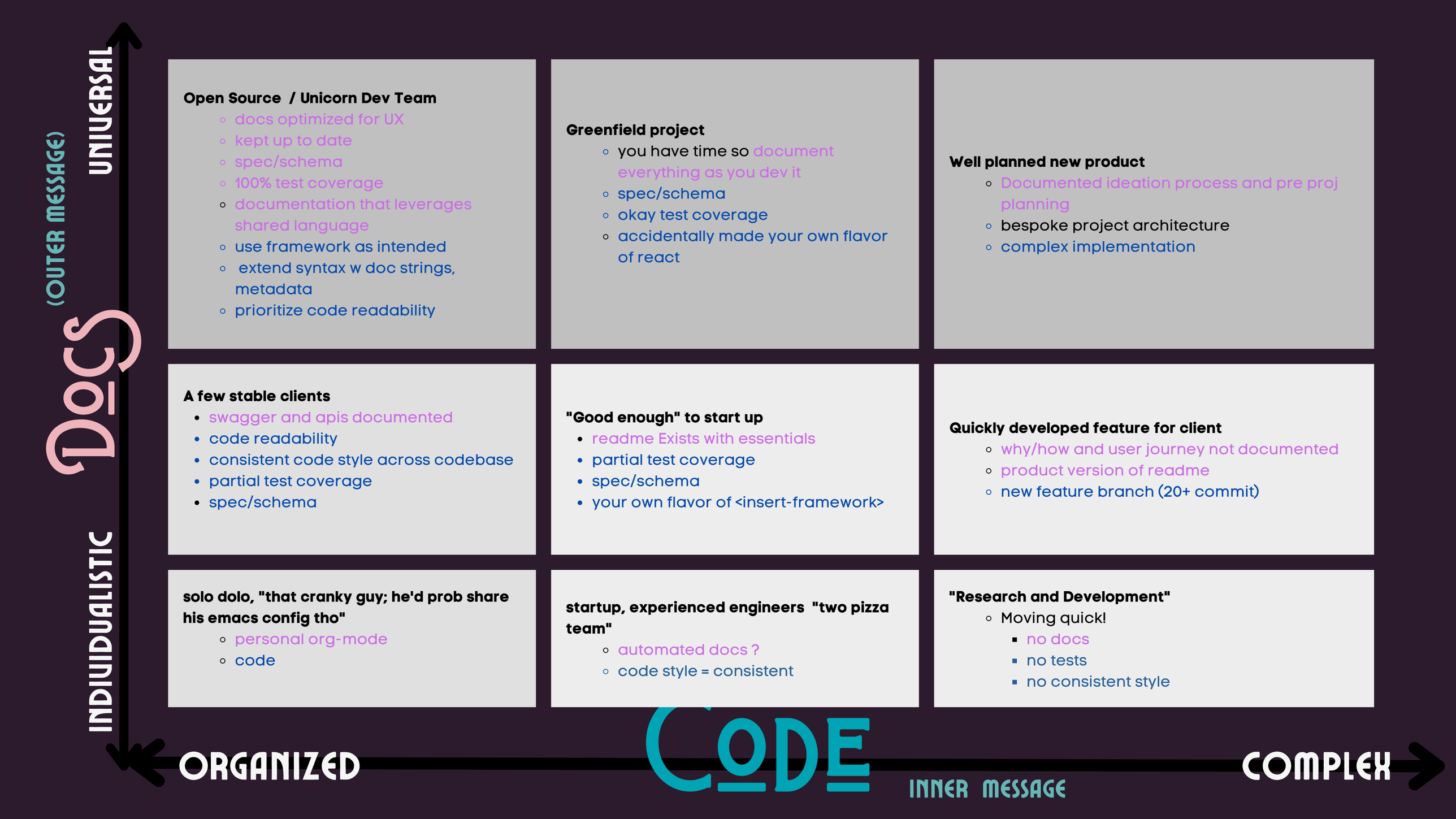
ORGANIZED

CODE INNER MESSAGE

COMPLEX


Open Source / Unicorn Dev Team
docs optimized for UX
kept up to date
spec/schema
100% test coverage
documentation that leverages shared language
use framework as intended
extend syntax w doc strings, metadata
prioritize code readability
"Good enough" to start up
readme Exists with essentials
partial test coverage
spec/schema/types
your own flavor of <insert-framework>
UNIVERSAL
DOCS
(OUTER MESSAGE)
INDIVIDUALISTIC
ORGANIZED
CODE
INNER MESSAGE
COMPLEX


Open Source / Unicorn Dev Team
docs optimized for UX
kept up to date
spec/schema
100% test coverage
documentation that leverages shared language
use framework as intended
extend syntax w doc strings, metadata
prioritize code readability
"Good enough" to start up
readme Exists with essentials
partial test coverage
spec/schema
your own flavor of <insert-framework>
"Research and Development"
no docs
no tests
no consistent style
UNIVERSAL
DOCS
(OUTER MESSAGE)
INDIVIDUALISTIC
ORGANIZED
CODE
INNER MESSAGE
COMPLEX


DOCS
(OUTER MESSAGE)
UNIVERSAL
INDIVIDUALISTIC
Open Source / Unicorn Dev Team
docs optimized for UX
kept up to date
spec/schema
100% test coverage
documentation that leverages shared language
use framework as intended
extend syntax w doc strings, metadata
prioritize code readability
Greenfield project
you have time so document everything as you dev it
spec/schema
okay test coverage
accidentally made your own flavor of react
Well planned new product
Documented ideation process and pre proj planning
bespoke project architecture
complex implementation
A few stable clients
swagger and apis documented
code readability
consistent code style across codebase
partial test coverage
spec/schema
"Good enough" to start up
readme Exists with essentials
partial test coverage
spec/schema
your own flavor of <insert-framework>
Quickly developed feature for client
why/how and user journey not documented
product version of readme
new feature branch (20+ commit)
solo dolo, "that cranky guy; he'd prob share his emacs config tho"
personal org-mode
code
startup, experienced engineers "two pizza team"
automated docs ?
code style = consistent
"Research and Development"
Moving quick!
no docs
no tests
no consistent style
ORGANIZED
CODE
INNER MESSAGE
COMPLEX

DOCS

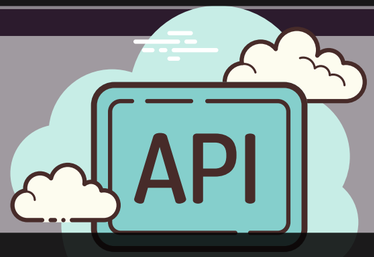
GOOD

EVIL

LAWFUL GOOD

NEUTRAL GOOD

CHAOTIC GOOD



LAWFUL NEUTRAL

TRUE NEUTRAL

CHAOTIC NEUTRAL

LAWFUL EVIL

NEUTRAL EVIL

CHAOTIC EVIL

LAWFUL

CODE

CHAOTIC

Sources

@lambduhh